Textual Entailment Through Extended Lexical Overlap

Rod Adams
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, Texas
rod@hlt.utdallas.edu

Abstract

This paper presents a system of textual entailment based primarily on the concept of lexical overlap. The system begins with a bag of words similarity overlap measure, derived from a combination of WordNet lexical chains to form a mapping of terms in the hypothesis to the source text. It then looks for negations not found in the mapping, and for the lexical edit distance of the mapping. These items are then entered into a decision tree to determine the overall entailment.

1 Introduction

Textual entailment is the task of taking a pair of passages, referred to as the *text* and the *hypothesis*, and labeling whether or not the hypothesis (H) can be fully inferred from the text (T), as is illustrated in Pair 1. In Pair 1, the knowledge that an attorney representing someone's interests entails that they work for that person.

Pair 1 (Dev IE 58)

T: "A force majeure is an act of God," said attorney Phil Wittmann, who represents the New Orleans Saints and owner Tom Benson's local interests.
H: Phil Wittmann works for Tom Benson.

The Second PASCAL Recognizing Textual Entailment Challenge[1] is an evaluation forum focusing on the test of automated methods of determining entailment. In the challenge, there are two corpora, each consisting of 800 annotated pairs of texts and hypotheses. Pairs are annotated as to whether there exists a positive entailment between them and from which application domain each example came from. The domains, or tasks, used in this challenge were: Question Answering (QA), Information Retrieval (IR), Information Extraction (IE), and Multi-Document Summarization (SUM). Instances were distributed evenly amongst the four tasks in both corpora, as were the number of positive and negative examples. One corpus was designated for development and training, while the other was reserved for testing.

In the first PASCAL RTE Challenge (Dagan et al., 2005), one of the best performing submissions was (Glickman et al., 2005), which determined a probability of entailment by comparing the words of the text and hypothesis using the results from an Internet search engine.

The goal of the system presented here is to build upon (Glickman et al., 2005)'s efforts to improve accuracy. The decision to concentrate on strict lexical methods was made so that the system could remain relatively simple and be easily applied to various entailment applications. Ideally, the system could serve as a baseline for evaluating future systems, or as a component inside a more complex system.

This paper is organized as follows: The design and implementation are discussed in Section 2, followed by an analysis of the results in Section 3. Afterwards, there is a comparison of the two corpora in Section 4, and concludes with suggestions for future efforts in Section 5.

[1] http://www.pascal-network.org/Challenges/RTE2/

```
(Phil, Phil)[1.0]
(Wittmann, Wittmann)[1.0]
(works, acts)[0.9]
(Tom, Tom)[1.0]
(Benson, Benson)[1.0]
```

Figure 1: Token Map for Pair 1 (Dev IE 58)

2 Framework

This system follows a four step framework. The first step is a tokenization process that applies to the content words of the text and hypothesis. The second step is building a "token map" of how the individual tokens in the hypothesis are tied to those in the text, as explained in Section 2.1. Thirdly, several features, as described in Section 2.2 are extracted from the token map. Finally, the extracted features are fed into Weka's (Witten and Frank, 2005) J48 decision tree for training and evaluation.

2.1 The Token Map

Central to this system is the concept of the token map. This map is inspired by (Glickman et al., 2005)'s use of the most probable lexical entailment for each hypothesis pair, but has been modified in how each pair is evaluated, and that the mapping is stored for additional extraction of features. The complete mapping is a list of (H_i, T_j) mappings, where H_i represents the i^{th} token in the hypothesis, and T_j is similarly the j^{th} token in the text. Each mapping has an associated similarity score. There is one mapping per token in the hypothesis. Text tokens are allowed to appear in multiple mappings.

The mappings are created by considering each hypothesis token and comparing it to each token in the text and keeping the one with the highest similarity score. Figure 1 illustrates the mapping for Pair 1.

Similarity Scores A similarity score ranging from 0.0 to 1.0 is computed for any two tokens via a combination of two scores. This score can be thought of as the probability that the text token implies the hypothesis one, even though the methods used to produce it were not strictly probabilistic in nature.

The first score is derived from the cost of a WordNet (Fellbaum, 1998) lexical chain. The lexical chains between two tokens are built with the method reported in (Hirst and St-Onge, 1998), and designated as $Sim_{WN}(H_i, T_j)$. Exact word matches are always given a score of 1.0, words that are morphologically related or that share a common sense are 0.9 and other chains give lower scores down to 0.0. This method of lexical chaining makes use of three groups of WordNet relations: *Up* (e.g. hypernym, member meronym), *Down* (e.g. hyponym, cause) and *Horizontal* (e.g. nominalization, derivation). The chain can follow only certain combinations of these groupings, and assigns penalties for each link in the chain, as well as for changing from one direction group to another. An example chain from Pair 4 is detailed in Figure 2

```
held→hold.v#6
    →[Down/hyponym] keep.v#3
    →[Down/hyponym] store.v#2
    →[Down/hyponym] tank.v#1
    →tanks
```

$$\begin{aligned} &Sim_{WN}(\texttt{tanks}, \texttt{held}) \\ &\quad = 0.8 - 0.1 \cdot (length + dir\_changes) \\ &\quad = 0.8 - 0.1 \cdot (3 + 0) \\ &\quad = 0.5 \end{aligned}$$

Figure 2: Sample Lexical Chain from Pair 4 (Dev SUM 330)

The secondary scoring routine is the lexical entailment probability, $lep(u, v)$, from (Glickman et al., 2005). This probability is estimated by taking the page counts returned from the *AltaVista*[2] search engine for a combined u and v search term, and dividing by the count for just the v term. This can be precisely expressed as:

$$\mathrm{Sim}_{AV}(H_i, T_j) = \frac{\mathrm{AVCount}(H_i \,\&\, T_j)}{\mathrm{AVCount}(T_j)}$$

The two scores are combined such that the secondary score can take up whatever slack the dominant score leaves available. The exact combination is:

$$\begin{aligned} \mathrm{Sim}(H_i, T_j) = {} & \mathrm{Sim}_{WN}(H_h, T_t) \\ & + \alpha \cdot (1 - \mathrm{Sim}_{WN}(H_h, T_t)) \cdot \mathrm{Sim}_{AV}(H_h, T_t) \end{aligned}$$

where α is a tuned constant ($\alpha \in [0, 1]$). Empirical analysis found the best results with very low values

[2] http://www.av.com

of α^3. This particular combination was chosen over a strict linear combination, so as to more strongly relate to Sim_{WN} when it's values are high, but allow Sim_{AV} to play a larger role when Sim_{WN} is low.

2.2 Feature Extraction

The following three features were constructed from the token map for use in the training of the decision tree, and producing entailment predictions.

Baseline Score This score is the product of the similarities of the mapped pairs, and is an extension of (Glickman et al., 2005)'s notion of $P(H|T)$. This is the base feature of entailment.

$$Score_{BASE} = \prod_{(H_i,T_j) \in Map} \text{Sim}(H_i, T_J)$$

One notable characteristic of this feature is that the overall score can be no higher than the lowest score of any single mapping. The failure to locate a strong similarity for even one token will produce a very low base score.

Unmapped Negations A token is considered unmapped if it does not appear in any pair of the token map, or if the score associated with that mapping is zero. A token is considered a negation if it is in a set list of terms such as `no` or `not`. Both the text and the hypothesis are searched for unmapped negations, and total count of them is kept, with the objective of determining whether there is an odd or even number of them. A (possibly) modified, or flipped, score feature is generated:

$n =$ # of negations found.

$$Score_{NEG} = \begin{cases} Score_{BASE} & \text{if } n \text{ is even,} \\ 1 - Score_{BASE} & \text{if } n \text{ is odd.} \end{cases}$$

Lexical Edit Distance The goal of this feature is to measure how clustered the mapped tokens in the text are. It counts any unmapped tokens which appear as gaps between two mapped tokens. For example, Pair 2 has four such gap tokens (`will`, `PeopleSoft`, `JD`, and `Edwards`). This count is scaled to the length of the hypothesis.

Development Corpus

Feature Set	IE	IR	QA	SUM	All
Base	53.0	60.0	54.0	78.5	61.4
Base + Neg	53.0	60.0	54.0	78.5	61.4
Base + Dist	62.0	63.0	66.5	73.5	66.3
Base + Task	53.0	60.0	66.0	78.5	64.4
Base + Neg + Dist	62.0	63.0	66.5	73.5	66.3
Base + Neg + Task	53.0	60.0	69.0	78.5	65.1
Base + Dist + Task	63.5	60.0	70.0	78.5	68.0
All Features	63.5	60.0	70.5	78.5	68.1

Table 1: Accuracy by task and feature set, when trained and tested on the "development" corpus

Test Corpus

Feature Set	IE	IR	QA	SUM	All
Base	52.0	59.5	49.5	72.0	58.3
Base + Neg	52.0	59.5	49.5	72.0	58.3
Base + Dist	50.0	51.5	66.5	66.5	58.6
Base + Task	52.0	59.5	54.5	72.0	59.5
Base + Neg + Dist	50.0	51.5	66.5	66.0	58.5
Base + Neg + Task	52.0	59.5	55.5	72.0	59.8
Base + Dist + Task	50.5	59.5	69.0	72.0	62.8
All Features	50.5	59.5	68.5	72.0	62.6

Table 2: Accuracy by task and feature set, when trained on the "development" corpus, and tested on the "test" corpus

Pair 2 (Dev QA 638)

T: And, despite its own suggestions to the contrary, Oracle will sell PeopleSoft and JD Edwards financial software through reseller channels to new customers.
H: Oracle sells financial software.

Task The task domain used for evaluating entailment (i.e. IE, IR, QA or SUM) was also used as a feature to allow different thresholds among the domains.

3 Results

The accuracy of the system on the Pascal RTE II corpora is summarized in Tables 1 and 2. Confidence scores can be taken from the probabilities in the decision tree, but offer insignificant gains and will not be discussed here. Overall, this system achieves an accuracy of 62.8%. Detailed analysis of results on a per feature basis is given below.

3.1 Feature Performance

Baseline The baseline entailment feature gives overall accuracies of 61.4% and 58.3% in the two corpora. The feature performs better at the Summarization task and worse in the IE and QA tasks.

[3] The results reported here used $\alpha = 0.1$

For its best two tasks (IR and SUM), the addition of other features gives no performance gain.

This feature performs exceptionally well when there are tokens in the hypothesis that are mapped either very poorly or not at all, as in (Pair 3), where the low scores for mapped pairs (`MTV`, `rock`) [0.008] and (`moonmen`, `rock`) [<0.001], among other poor mappings, correctly indicates a negative entailment.

Pair 3 (Dev SUM 286)

T: Green Day, who arrived at the venue in the vintage green convertible from their gritty "Boulevard of Broken Dreams" video, won best rock video and video of the year for the clip and two of their leading eight nominations.
H: Rock was resplendent at the MTV Video Music Awards on Sunday night, as the veteran punk group, Green Day, took home seven moonmen.

This feature is prone to false positives, such as (Pair 4), where the score should be relatively low, since the text has nothing to do with tanks. However, the system finds a mapping of (`tanks`, `held`)[0.502[4]], keeping the overall baseline score relatively high.

Pair 4 (Dev SUM 330)

T: Two British soldiers have been arrested in the southern Iraq city of Basra, sparking clashes outside a police station where they are being held.
H: Two British tanks, sent to the police station where the soldiers are being held, were set alight in clashes.

Unmapped Negations This feature allows the system to correctly determine that (Pair 5) is a false entailment, but for the wrong reasons. It found the `nor` in the text, and thus flips the baseline score, setting the entire entailment to false, whereas the true reason for the false entailment is that text is discussing a future probability, not current fact.

Pair 5 (Dev QA 230)

T: Nor is it clear whether any US support to Germany, in favour of Bonn as the WTO headquarters, would necessarily tilt a decision in that direction.
H: The WTO headquarters is in Bonn.

Whatever questionable gains this feature has is offset by cases such as (Pair 6), where the unrelated `not` incorrectly forced the overall prediction to incorrectly be false.

Pair 6 (Dev QA 101)

T: Vatican spokesman Joaquin Navarro-Valls announced that Pope John Paul was still in serious condition, but he was not in a coma.
H: Joaquin Navarro-Valls is the Vatican Spokesman.

Overall, this feature is too crude to offer any benefit to the system as a whole in its current form. Several refinements were tested, but none produced suitable gains.

Lexical Edit Distance The edit distance feature performs very differently between the two corpora. In the development corpus, it provides an almost 5% gain over baseline and a 3% gain with all features. Its advantages are most pronounced in the IE task, where gains around 10% are seen. For example, in (Pair 7), the mapping is $\{$(`Washington`, `Washington`) [1.000], (`London`, `London`) [1.000], (`part`, `met`) [0.440][5]$\}$, accounting for ten gaps in a three token long hypothesis. This relatively extreme gap count allows the system to correctly mark the pair as having a false entailment.

Pair 7 (Dev IE 12)

T: He met U.S. President, George W. Bush, in Washington and British Prime Minister, Tony Blair, in London.
H: Washington is part of London.

This feature performs quite differently in the test corpus, where it generally accounts for no gain overall, and even hampers the accuracy in places. For example, (Pair 8) has the exact match mapping of the tokens `Mauricio`, `Pineda`, `killed` and `Morazan`. However, the intervening apposition 'a sound technician for the local canal Doce television station', generates a relatively high gap count, which forces the system to incorrectly mark the pair as having a false entailment.

Pair 8 (Test IE 84)

T: Salvadoran reporter Mauricio Pineda, a sound technician for the local canal Doce television station, was shot and killed today in Morazan department in the eastern part of the country.
H: Mauricio Pineda was killed in Morazan.

Of interest is that this reversal of benefit does not present itself in the QA task. Despite the baseline

[4] See Figure 2, and then combine with $Sim_{AV}(\texttt{tanks}, \texttt{held})$.

[5] `part` → `change` → `have` → `meet`

Corpus	IE	IR	QA	SUM	All
Development	55.5	59.5	68.5	76.0	64.9
Test	51.5	64.0	68.5	73.5	64.4
Combined	56.8	64.5	66.3	75.3	65.7

Table 3: Leave-One-Out Accuracy

feature performing weaker with the test corpus than with the development corpus, adding just the edit distance feature brings the test corpus accuracy up to development corpus levels in that task.

Task Allowing the decision tree to consider the entailment task affords consistent gains of 2-4$\%$ in accuracy.

4 Comparison of the Corpora

In an attempt to quantify any differences between the two corpora, cross validation tests were performed on each corpus separately, and then with the two combined together. Give the relatively small size of the corpora, a leave one out[6] system was chosen. These results are summarized in Table 3.

If the two corpora were similar in character, one could assume that the results for the two to be similar to each other, as well as to the combined corpus test. Three of the four tasks show variances of over 3% between the two corpora, with IR showing a change of 4.5%. Results from the QA task, which alone kept the same performance between the two corpora, become surprising when compared to the results from the combined corpus, where it showed a 2% drop in performance.

However, it is unclear at this time if these differences in performance are due to actual differences in the corpora, or if they are merely artifacts of this system.

5 Future Directions

Several avenues of improvement to this system are possible.

Lexical Chains Besides (Hirst and St-Onge, 1998), there are several other methods for computing term similarity through lexical chains. A reasonable comparison of techniques can be found in (Budanitsky and Hirst, 2001), where for their purposes they found (Jiang and Conrath, 1997) and (Lin, 1998b) to be strong measures.

Web Based Similarity As an alternative to the simple web search page counts used in this and other systems, it is worth considering approaches which build a token similarity measure from the Internet, such as (Lin, 1998a), or IR techniques such as (Baeza-Yates and Ribeiro-Neto, 1999).

Apposition Detection Enhancing the edit distance metric to be able to discount appositions in its counting of gaps could help resolve the false negatives this phenomenon produces.

6 Conclusions

This paper presented a system which predicts the textual entailment between two passages, with accuracy better than the baseline of always choosing `true` or `false` [7]. True to its goals, the system remained quite simple, with the only external dependency being the nearly ubiquitous WordNet.

7 Acknowledgments

I would like thank my adviser, Dr. Harabagiu and the various anonymous reviewers who helped in these endeavors.

References

Ricardo Baeza-Yates and Berthier Ribeiro-Neto, 1999. *Modern Information Retrieval*, pages 131–134. Addison Wesley.

Alexander Budanitsky and Graeme Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Second meeting of the North American Chapter of the Association for Computational Linguistics*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognizing textual entailment challenge. In *PASCAL RTE Challenge*.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. Web based probabilistic textual entailment. In *PASCAL RTE Challenge*.

[6] Leave one out is a strategy such that if there are n items in the corpus, it is equivalent to n-fold cross validation. Thus, 800-fold validation was done for the Development and Test corpora, and 1600-fold validation was done for the combined corpus.

[7] Since both are present in equal numbers, either system would yield 50% accuracy.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database*, pages 305–332. The MIT Press.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference on Research in Computational Linguistics*.

Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *COLING-ACL*.

Dekang Lin. 1998b. An information-theoretic definition of similarity. In *International Conference of Machine Learning*.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition.